Une image contenant texte

Description générée automatiquement

Département de génie informatique et génie logiciel

Travail pratique 1

Plan et test, Maintenabilité et Fiabilité

**Présenté à :**

Rhouma Naceur

**Dans le cadre du cours :**

LOG8371 – Ingénierie de la qualité en logiciel

**Groupe : 01**

**Soumis par :**

Minh-Tri Do –

Jeremy Hage – 2014233

Jonathan Siclait – 2026416

Le 2 février 2022

Table des matières

**Plan d’assurance qualité**

Description du logiciel

AntennaPod est une application de gestion de podcast fondée en 2011 par Daniel Oeh. Cette application Android, codée en Java, est ouverte au public et est complètement libre de droits. Elle a été développée sans but lucratif dans le but de permettre à n’importe qui de l’utiliser afin d’écouter une multitude de podcasts. De plus, puisque les déeveloppeurs ne recherchent pas d’argent, il n’y a aussi aucune publicité dans ce système. Elle a présentement été installée sur plus de 180 000 appareils. C’est donc sur cette application que nous avons décidé d’effectuer cette analyse. Dans le cadre de ce rapport, l’analyse se penchera sur les modules « app » et « core ». [[1]](#footnote-1)[[2]](#footnote-2)

Importance de la qualité

La qualité est particulièrement du code est particulièrement importante dans le cas de AntennaPod car ce logiciel est ouvert au public (open source). En effet, le fait que n’importe qui dans le monde peut ajouter des composantes au projet sans qu’il n’y aie une supervision associée rend la présence de standards de codage, de vérification de code et de tests multiples essentielle pour assurer une bonne maintenabilité du code ainsi qu’une application sécuritaire pour les utilisateurs.

Parties prenantes

En analysant le projet AntennaPod, il est facile de remarquer qu’il existe 3 principales parties prenantes. Il existe tout d’abord l’audience. C’est évidemment dans intéressant pour les utilisateurs de l’application qu’elle fonctionne. Ensuite, il y a les créateurs de podcasts qui, eux, placent leurs podcasts dans l’application dans le but de maximiser le nombre de gens qui peuvent l’écouter. Enfin, il existe aussi les développeurs d’AntennaPod qui tiennent à ce que l’application fonctionne car ils ont travailler sur ce projet à but non lucratif afin de donner la meilleure expérience utilisateur possible.

Couverture de qualité

|  |  |  |  |
| --- | --- | --- | --- |
| **Critère** | **Sous-critère** | **Objectif** | **Mesure de validation** |
| Fonctionnalité | Complétude : Critère qui décrit le fait que chaque donnée liée à un utilisateur est bien remplie | Amener le pourcentage P de qualité de données en lien avec la complétude des données à 100%. | Formule : P = A/B où A est le nombre de données présentes et B est le nombre de données requises |
| Conformité : Critère qui décrit le degré auquel les données respectent les standards, règles et conventions en vigueur en lien avec la qualité de données. | Amener le pourcentage P de conformité des données à 100%. | Formule : P = A/B où A est le nombre de données conformes aux réglementations et B est le nombre de données totales. |
| Précision : Critère qui décrit le degré auquel les données des utilisateurs contiennent des attributs qui discriminent les bonnes données dans un contexte spécifique. | Amener le pourcentage P de précision des attributs dans la base donnée à 100%. | Formule : P = A/B où A est le nombre de valeurs ayant la précision requise |
| Fiabilité | Disponibilité : Critère qui décrit le temps que entre les pannes de l’application | Amener le pourcentage P de temps entre les pannes à 99.5% (30 minutes par semaine). | Formule : P = A/B où A est le temps entre les pannes et B est le temps total |
| Tolérance aux pannes : Critère qui décrit le pourcentage du système qui fonctionne lors d’une panne | S’assurer que le pourcentage du système qui reste fonctionnel en cas de panne est de 80%. | Formule : P = A/B où A est le pourcentage de fonctionnalités fonctionnelles en cas de panne et B est le nombre total de fonctionnalités. |
| Récupérabilité : Critère qui décrit le degré auquel le système peut fonctionner dans le cas d’erreur ou de « crash ». | Amener le pourcentage P de données perdues lors d’une erreur à moins de 1%. | Formule : P = A/B où A est le nombre de données perdues lors d’un « crash » et B le nombre de données total. |
| Maintenabilité | Modifiabilité : Critère qui décrit le degré auquel le système peut être facilement modifié | Réduire le pourcentage P de fichiers nécessaires à modifier lors d’une seule modification à 2 ou moins. | Formule : P = A/B où A est le nombre de fichiers modifiés par modification et B est le nombre de modifications. |
| Testabilité : Critère qui décrit le degré auquel les composants de l’application peuvent être testés individuellement avec facilité | Amener le pourcentage P de focntions autonomes à 70%. | Formule : P = A/B où A est le nombre de fonctions autonomes et B est le nombre de fonctions. |
| Réutilisabilité : Critère qui décrit le nombre de réutilisation des fonctions de l’application | Avoir un nombre moyen P de fonctions de réutilisés plus grand que 2. | Formule : P = A/B où A est le nombre de fois que chaque fonction est utilisée et B est le nombre de fonctions. |

Stratégie de validation

Il y a plusieurs stratégies de validation qui peuvent et doivent être utilisées dans un projet de cette envergure. Une première stratégie, la moins formelle est la revue personnelle. Elle consiste tout simplement en un processus effectué tout au long du développement où le développeur prend des pauses de développement afin d’effectuer des « checklists » de points à réviser dans son code afin de s’assurer du bon fonctionnement de celui-ci. Il utilise ensuite ces listes afin de faire une révision pour minimiser le nombre de bugs et problèmes qui pourraient survenir.

Une autre stratégie qui peut être utile pour AntennaPod serait les revues par les pairs. Plus précisément, un walk-through serait la stratégie optimale de validation car, dans un contexte ouvert au public comme celui-ci, les pairs vont remarquer des problèmes et laisser des commentaires pour le développeur qui a codé la fonctionnalité en question. Les développeurs qui effectueront les revues devront donc présenter à l’auteur de la fonctionnalité une présentation détaillée de leur revue. Ce type de revue pourra aider à l’analyse de la maintenabilité du projet puisque les pairs analyseront, en partie, la maintenabilité du code.

1. <https://antennapod.org/about/> [↑](#footnote-ref-1)
2. <https://antennapod.org/> [↑](#footnote-ref-2)